Jus Mundi at SemEval-2023 Task 6: Using a Frustratingly Easy Domain Adaption for a Legal Named Entity Recognition System

Luis Adrián Cabrera-Diego and **Akshita Gheewala**
Jus Mundi / 30 rue de Lisbonne, 75008, Paris, France
{a.cabrera,a.gheewala}@jusmundi.com

Abstract

In this work, we present a Named Entity Recognition (NER) system that was trained using a Frustratingly Easy Domain Adaptation (FEDA) over multiple legal corpora. The goal was to create a NER capable of detecting 14 types of legal named entities in Indian judgments. Besides the FEDA architecture, we explored a method based on overlapping context and averaging tensors to process long input texts, which can be beneficial when processing legal documents. The proposed NER reached an F1-score of 0.9007 in the sub-task B of Semeval-2023 Task 6, Understanding Legal Texts.

1 Introduction

In this paper, we present *Jus Mundi's*[1] participation in Task 6 of Semeval-2023, *LegalEval: Understanding Legal Texts* (Modi et al., 2023). Specifically, Jus Mundi participated in the sub-task B, *Legal Named Entities Extraction*, which consisted of creating a Named Entity Recognition (NER) system for Indian judgment documents in English. The goal was to predict multiple types of entities from both the judgment preamble and the judgment body. Our participation consisted of a model trained using a *Frustratingly Easy Domain Adaptation* (FEDA) algorithm (Daumé III, 2007; Kim et al., 2016; Cabrera-Diego et al., 2021b).

The FEDA algorithm, as its name indicates, is a type of domain adaption method created by Daumé III (2007). The objective of using a FEDA algorithm is to learn common and domain-specific patterns between multiple datasets (Daumé III, 2007), but it can be used also to mix multiple datasets despite not having the same tagset, as seen in Cabrera-Diego et al. (2021b). Moreover, a FEDA algorithm simplifies many of the questions that generally arise while exploring other types of domain adaptation and transfer learning techniques. For instance, with a FEDA algorithm, it is not necessary to determine which layers should be frozen, fine-tuned, or substituted.

The rest of the paper is organized as follows. In Section 2, we introduce the background for the proposed work. This is followed by the methodology in Section 3. The data and the experimental settings are described in Section 4 and Section 5, respectively. In Section 6, we present the results and discuss them. Finally, the conclusions and future work are detailed in Section 7.

2 Related Work

Named entity recognition (NER) is a fundamental Natural Language Processing (NLP) task that consists of identifying entities that semantically represent elements such as locations, organizations, and people (Li et al., 2020).

While the applications and advantages of using NER systems are well known, in many domains and languages, it is a task that continues to be a challenge. This is the case within the legal domain, where the extent of research regarding NER on this subject is still small in contrast to more general domains, such as CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) or OntoNotes v5 (Ralph Weischedel et al., 2011). The main reasons are the lack of annotated data to train NER systems (Pilán et al., 2022) but also the complexity of processing legal documents (Gupta et al., 2018), e.g. long sentences and specialized vocabulary.

Nonetheless, in the following paragraphs, we present some of the most representative works regarding the detection of legal named entities.

For Indian documents, we can highlight the works of Gupta et al. (2018) and Kalamkar et al. (2022). In the former work, the authors trained an NER using OntoNotes v5 (Ralph Weischedel et al., 2011) and applied it to legal documents in a zero-shot approach. In the latter, the authors proposed an NER corpus from different Indian court

[1] https://jusmundi.com

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1783–1790

judgments. Moreover, they trained an NER system using a Transformer-based architecture (Vaswani et al., 2017) and make use of coreference resolution and rules to reconcile named entities.

In the work of Barriere and Fouret (2019), the authors created an NER robust to spelling mistakes for French court decisions using deep learning and contextual dictionaries of entities. For Romanian and German, we have the works of Păiș et al. (2021) and Leitner et al. (2019), respectively. Both works created their legal corpus and utilized a BiLSTM-CRF architecture to train an NER.

In the legal domain, it is frequent to find other NER systems in anonymization tools. For instance, Schamberger (2021) make use of a BiLSTM-CRF architecture with BERT (Devlin et al., 2019) embeddings to anonymize German court rulings. In Pilán et al. (2022), the authors create an NER for anonymizing documents from the European Court of Human Rights using a Transformer-based architecture. Similarly, in Oksanen et al. (2022) the authors use a transformer-based NER for anonymizing Finnish documents.

Regarding the FEDA algorithm, it was originally proposed by Daumé III (2007) for sparse machine learning algorithms. It is a simple method that consists of duplicating the input features. Then, Kim et al. (2016), proposed a neural network version of FEDA in which certain layers are activated according to their respective training dataset. Later, Cabrera-Diego et al. (2021b) used a FEDA algorithm to train multiple NER systems for less-resourced languages using BERT and multilingual datasets with different tagsets.

3 Methodology

We define our FEDA architecture as a collection of dense layers built over a pre-trained language model, which follows the same ideas presented by Cabrera-Diego et al. (2021b). In other words, it is composed of one general FEDA layer and multiple specialized FEDA layers; each specialized FEDA layer is connected to the general one. Nonetheless, we differ from Cabrera-Diego et al. (2021b) on how the FEDA layers are defined and connected, as we will describe in this section, but also on how we train the model, as it will be shown in Section 5.

Thus, let us establish a FEDA layer F as a GELU activation layer (Hendrycks and Gimpel, 2016) followed by a Linear one. We define a General FEDA layer G as the stack of two layers F joined by a

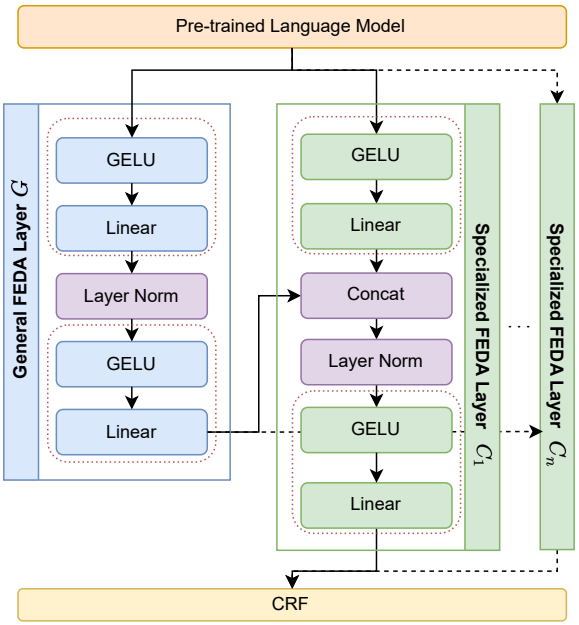

Figure 1: Diagram of the FEDA architecture used in this work.

Layer Normalization layer (Ba et al., 2016). And, we define a Specialized FEDA layer C_x as a stack of two layers F joined by a Concatenation layer and Layer Normalization layer. The G layer is linked to each C_x layer through their respective concatenation layer. Moreover, on top of all the Specialized C_x layers, we define a Conditional Random Field (CRF) (Lafferty et al., 2001) that processes their output, similar to the approach used by (Ma and Hovy, 2016) to improve an NER system.

For a given collection of training datasets defined as $\mathcal{D} = \{D_1, D_2, \ldots, D_n | n > 1\}$, where each D_x is composed of i entries, e.g. a paragraph or a sentence, of type e_x^i. Our FEDA architecture is composed of one General FEDA layer G and n Specialized FEDA layers C_x, such that we have $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$. We present the diagram of the proposed FEDA architecture in Figure 1.

For each entry e_x^i, the output created by the language model is introduced into G and its corresponding C_x layer. Then, the output generated by G is introduced into C_x by concatenating the tensor to the output of the first layer F in the specialized FEDA layer C_x. The concatenated output is then passed through the rest of the layer C_x. The output of C_x then goes into the CRF layer.

In summary, G represents a classifier that receives an input comprising of all the entries e^i from all the datasets in $\mathcal{D}$, while $C_x \in \{\mathcal{C} | 0 < x \leq n\}$ represents a specialized classifier that focuses only on the entries e_x^i that belong to the dataset $D_x \in$

$\{\mathcal{D}|0 < x \leq n\}$. The CRF layer is shared by all the datasets D_x.

4 Data

As we make use of a FEDA architecture, we used five different datasets related to the legal domain for training our submitted models.

The main training corpus, henceforth LegalNER, is described in Kalamkar et al. (2022). It is composed of Indian court judgments that have been annotated using 14 types of named entities related to the legal domain: Case number, Court, Date, Geopolitical entity (GPE), Judge, Lawyer, Organization, Other person, Petitioner, Precedent, Provision, Respondent, Statute and Witness.[2] Furthermore, this corpus covers two different parts of a court judgment, the preamble, and the judgment body. The former is a type of introductory cover document where certain legal elements are presented in a concise way without necessarily using sentences or phrases. The latter is a paragraph or a sentence found in the judgment body.

Granted that LegalNER is split into Train, Development, and Test partitions, we created our own development partition due to two reasons. Firstly, the development corpus was provided after the competition started. Moreover, we discovered that the development partition was not proportionally balanced with the training corpus in terms of the quantity and variety of entities. Therefore, we created our own development corpus from the training partition. We used a stratified approach, in which we considered three aspects to create the development partition. The first aspect was related to cultural elements that could play a role in the quality of the NER system. Specifically, we classified the documents coming from various cities, states, and union territories into three groups: North and West, South and Center, and North East. Secondly, the proportion of entities had to be around 10% of the training set; this also included the proportion of groups previously defined. The third aspect was prioritizing unique courts for the development corpus. Consequently, this would allow us to finely evaluate the generalization of our NER system towards unseen courts. We included the original development corpus as part of the training partition once it was made available to the participants.[3] We present in Table 1, the statistics of each corpus partition.

We automatically pre-processed all the named entities in the train and development corpus partitions, to remove some undesirable patterns that were being learned by the model during the first experiments due to noise in the data. For instance, certain entities had trailing spaces or commas, but also honorifics, such as *Sri* and *Judge*. Specifically, we used a combination of regular expressions that matched the noisy patterns and rules to clean the data. Also, we manually added and/or changed certain entities in the training and development corpus after doing an error analysis on the predictions produced on the development corpus during the first experiments. For instance, in the document "*But it had then discovered that the real accused were Intertek and Prasanna Ghotage, who had together conspired to defraud not only DLC, but also SMC in the transaction.*"[4], only two entities were marked, *Intertek* and *Prasanna Ghotage*, both as Other Person. We changed *Intertek* to Organization, and we added *DLC* and *SMC* as Organizations. The pre-processing allowed us to have a cleaner and more standardized corpus.

The additional datasets are described as follows:

Edgar-NER: A legal corpus created by Au et al. (2022) and composed of 52 documents related to financial filings submitted by companies to the US Securities and Exchange Commission. It consists of 7 types of named entities: Location, Person, Business, Government, Court, Legislation/Act, and Miscellaneous.

Citations: A Jus Mundi corpus where different paragraphs from legal documents have been annotated with citations. In this corpus, the definition of citations covers all those defined in Kalamkar et al. (2022) as Precedents, Statue, Cases numbers, and Provisions.

Persons and Honorifics: This is also a Jus Mundi legal corpus that has been annotated with 2 labels, Person names, and Honorifics, such as *Mrs.* and *Q.C.*

Metadata: An internal Jus Mundi corpus that consists of automatically annotated cover pages of arbitration-related documents from different jurisdictions. Nine different types of labels exist, Arbitrator, Date, Claimant, Respondent, Expert, Institution, Lawyer, Title, and Member of the Tribunal.

[2] See Kalamkar et al. (2022) for a detailed description of the corpus and its entities.

[3] It was not considered for our development corpus. In summary, for training our final models we used the Orginal Train - Our Developement + Original Development partitions.

[4] ID `4cf4617887134f7d8aa83f1c3bdc613b` in the train judgment corpus.

Label	Corpus partition					
	Original Train		Original Development		Our Development	
	Preamble	Judgment	Preamble	Judgment	Preamble	Judgment
Court	1074	1293	118	178	145	112
Judge	1758 (-1, +8)	567 (+4)	166	8	170	41
Lawyer	3505 (+13)	0	589	0	358	0
Petitioner	2604 (-1)	464 (+3)	202	9	279	81
Respondent	3538 (-2)	324 (+1)	310	5	383	36
Case number	0	1040 (+2)	0	121	0	98
Date	0	1885 (+2)	0	222	0	185
GPE	0	1398 (-4, +6)	0	183	0	142
Organization	0	1441 (-1, +12)	0	159	0	98
Other Person	0	2653 (-9, +7)	0	276	0	310
Precedent	0	1351	0	177	0	112
Provision	0	2384	0	258	0	208
Statue	0	1804 (+2)	0	222	0	191
Witness	0	881 (+1)	0	58	0	137

Table 1: Statistics of the LegalNER corpus. The Original Train and Original Development partitions were given by the task organizers; between brackets, we indicate how many entities were manually removed or added by us. Our Development partition was sliced from the extended Original Train partition.

Despite being composed of around 9k documents, its coverage is not perfect due to OCR errors in addition to the process of automatic annotation using fuzzy matching.

5 Experimental Settings

All the named entities were encoded using BILOU (Beginning, Inside, Last, Outside/Other, Unique) labeling scheme, which has been precedented as an indirect way to improve NER predictions (Ratinov and Roth, 2009).

We use DeBERTa V3 (He et al., 2021) Large as a pre-trained language model. We trained multiple models during the evaluation period, for which the hyperparameters stayed the same, except for DeBERTa's sequence size. These are presented in Table 2.

During training, we also explored different ways to prevent overfitting. We found out that the best approach was to measure the overfitting per dataset. In effect, once a dataset started to overfit, it was dropped from the training. Followed by reloading the model's state on which we obtained the best performance of the dropped dataset and continuing the training. This differs from the approach followed by Cabrera-Diego et al. (2021b), on which the overfit was measured globally, denoting that certain datasets could affect the global performance.

Additionally, when an input text was longer than the maximum sequence size of DeBERTa, the input text was split into smaller sub-input texts with an overlapping context stride of size 40.[5] The overlapping context strides gives the NER system additional information about entities that can be found close or on the boundaries of a sub-input text. During prediction time, the tensors related to each sub-input text were concatenated before the CRF layer. The tensor portions belonging to the overlapping contexts were merged through an average. The above-described approach is shown in Figure 2. We decided to do an average of the overlapping context tensors because the output generated by the NER can be discordant depending on the context it is analyzed. For example, an entity might not exist in one sub-input text, it might be assigned a differ-

Hyperparameter	**Value**
Maximum Epochs	20
Early Stop Patience	2 to 5
Learning Rate	2×10^{-5}
Scheduler	Linear with warm-up
Warm-up Ratio	0.1
Optimizer	Lookahead (Zhang et al., 2019) AdamW with bias correction
AdamW ϵ	1×10^{-8}
Random Seed	12
Dropout rate	0.5
Weight decay	0.01
Clipping gradient norm	1.0
DeBERTa's sequence size	256, 384 and 512
Training Batch	32
Context stride	40

Table 2: Hyperparameters used for training the models.

[5] See Tokenizer documentation at https://huggingface.co

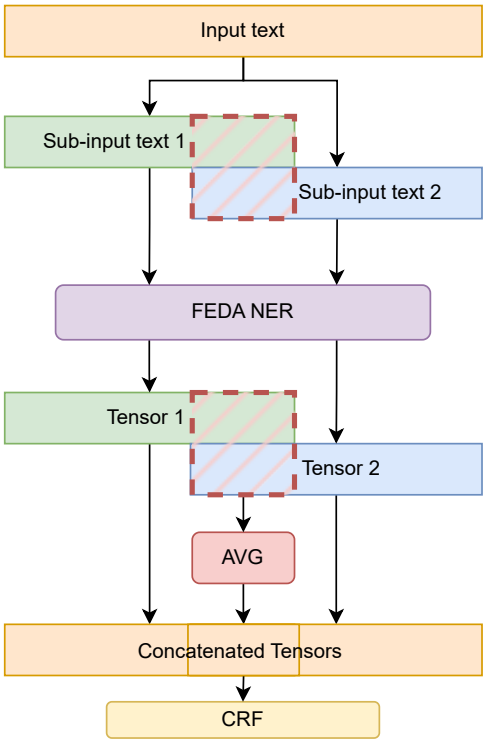

Figure 2: Approach used for predicting sentences longer than DeBERTa's sequence size at prediction time.

ent type, or it might comprise a different number of tokens. Thus, by averaging the tensors belonging to the overlapping contextual strides, we can simply and quickly represent a portion of text that was predicted twice by the NER, and give enough information about the tensors to the CRF. This method for dealing with long input texts is another extension to the approach undertaken in Cabrera-Diego et al. (2021b).

To improve the predictions, we also explored the merging of multiples models outputs through a simple majority voting system. In case of a tie in the voting, we gave priority to the model with the highest score obtained on the test partition.

Finally, we created a post-processing tool based on regular expressions that improved the entities to avoid some noise produced by the NER. For instance, in some entities, we closed unbalanced brackets while in others we removed leading and trailing punctuation marks, such as quotation marks. Some examples of entities that had to be post-processed: "*Section 147,148, 302/149 (*" to "*Section 147,148, 302/149*"; "*Rajendra Kumar (Verma*" to "*Rajendra Kumar (Verma)*".

6 Results and Discussions

Collectively, we submitted 7 different models to the evaluation platform of Task 6.[6] These were chosen based on their performance on our Development partition. The F1-score of the submitted models, in order of submission, were the following ones: 0.8853, 0.8861, 0.8949, 0.8908, 0.8953, 0.90071, and 0.90072.

The first model submitted, 0.8853, was produced by our baseline, a model that consisted of DeBERTa V3 + CRF layer. For the following models, we submitted only FEDA architectures. The last three submitted models used the majority voting approach and the last two applied the post-processing tool for cleaning the predictions. The use of the voting system allowed us to pass from an F1-score of 0.8949 to 0.8953, while the use of the post-processing filter surpassed an F1-score of 0.9007.

Bearing that the testing corpus has not been made public, a deep analysis of the results is unfeasible. Although in the following paragraphs, we present some aspects that we observed from the predictions obtained during the fine-tuning of our models and also, on the test submissions.

We noticed that pre-processing the training files played an important role in the training of the model. On the first trained models, before submission, we observed that certain noisy patterns were being learned, such as finishing entities with a comma. Nonetheless, some patterns could not be fixed because we did not know which was the correct one. For instance, some entities of type Date had the prefix *dt.* while some others did not.

It should be indicated that we explored the representation of newlines by adding a special character (`<NEWLINE>`) into DeBERTa's vocabulary. This was done following the works of Baldini Soares et al. (2019) and Cabrera-Diego et al. (2021a), in which special characters are used to make BERT (Devlin et al., 2019) focus on specific information. The goal was to try to introduce some kind of formatting to the NER, especially on the preambles, where the format is substantial to better determine an entity type. The performance gain of the models varied a lot. For instance, the third submitted model, with an F1-score of 0.8949, was one of the best models we trained. Nonetheless, other variations of our FEDA model and the special character `<NEWLINE>`, did not show to improve the results

[6] https://codalab.lisn.upsaclay.fr/competitions/9558

in the development partition. And thus, they were not submitted. The reason might be that LegalNER was the only corpus with this feature among all those used for training.

We discovered that the length of DeBERTa's sequence size affected the performance of the NER in great measure. This was especially noticeable in the preambles because it seems that to predict the type of certain entities correctly, a larger context was necessary. For instance, sometimes it was hard to determine whether a party in a preamble was a petitioner or a respondent after a long list of names.

Related to the previous aspect, sometimes our system had problems predicting a party correctly, either Petitioner or Respondent, when it was a person representing an organization. In this case, according to the guidelines, the named entity of the party had to include the full address. Subsequently, this was hard to train, not only because of the particularity of the guidelines but also because of the length of the context that had to be processed to predict the entity accurately.

Although we explored longer context strides as a countermeasure for long texts, we noticed that they did not improve the performance on the same scale that the stride was increased. This can be seen in our fourth submission, which achieved an F-score of 0.8908; we sent the best model and we only modified the size of the context stride to 150 characters. However, as it can be seen, its performance was not better than the third submission (0.8949).[7] One of the reasons could be that the approach used to merge overlapping contexts was not robust enough. This was more noticeable when the stride was larger than half of DeBERTa's sequence size. In other words, the average of the overlapping contexts became too noisy when an overlapping context was overlapping another context. In the future, we could try to train a dense layer or an attention layer to overcome this. However, these layers might need to be trained after the main NER model has been selected.

Additionally, we noticed that separating the early stop patience for each dataset, allowed us to improve the predictions up to a certain degree. As a result, the training of the models took longer and constrained us from exploring other aspects that could have improved the outcomes further. For instance, to increase the size of DeBERTa's sequence size to 512 for the final models.

Finally, we found out that on certain occasions the CRF could not correctly learn the tagset of the smallest training dataset explored, *Person and Honorifics*, after we separated the early stop patience. One of the reasons is that the CRF was shared among all the datasets and this particular dataset was very easy to train. This meant that the CRF had less time to understand the correct chain of possible types of entities, among all those existing. We theorize that this can be solved by creating a unique CRF for each dataset, which should also speed up the training of a model.

7 Conclusion

This work presented the participation of Jus Mundi at Semeval-2023 Task 6. Specifically, we proposed a Named Entity Recognition system that was trained using multiple legal datasets through Frustratingly Easy Domain Adaption (FEDA). The results showed us that our NER system is a performant tool on Indian judgments with an F1-score of 0.9007.

In the future, we intend to further explore the benefits and limitations of an NER based on a FEDA architecture. Moreover, we will probe different ways to speed up the training given that separating the early stop can make the training of the final models to be slower. We will also seek a better approach to join split sentences to predict entities more accurately in long documents, as commonly occurs in the legal domain. Finally, we will apply the proposed architecture to more complex tasks.

Acknowledgments

This work was possible thanks to the granted access of IDRIS (Institut du Développement et des Ressources en Informatique Scientifique) High-performance computing (HPC) resources under the allocation 2022-AD011012667R1 made by GENCI (Grand Équipement National de Calcul Intensif).

References

Ting Wai Terence Au, Vasileios Lampos, and Ingemar Cox. 2022. E-NER — An Annotated Named Entity Recognition Corpus of Legal Text. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 246–255, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

[7] On the development partition the score remained very similar, this is why the model with the modified stride was submitted.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Valentin Barriere and Amaury Fouret. 2019. May I Check Again? — A simple but efficient way to generate and use contextual dictionaries for Named Entity Recognition. Application to French Legal Texts. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 327–332, Turku, Finland. Linköping University Electronic Press.

Luis Adrián Cabrera-Diego, Jose G. Moreno, and Antoine Doucet. 2021a. Simple ways to improve NER in every language using markup. In *Proceedings of the 2nd International Workshop on Cross-lingual Event-centric Open Analytics (CLEOPATRA 2021)*, volume 2829, pages 17–31, Ljubljana, Slovenia. CEUR-WS.

Luis Adrián Cabrera-Diego, Jose G. Moreno, and Antoine Doucet. 2021b. Using a Frustratingly Easy Domain and Tagset Adaptation for Creating Slavic Named Entity Recognition Systems. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 98–104, Kiyv, Ukraine. Association for Computational Linguistics.

Hal Daumé III. 2007. Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ajay Gupta, Devendra Verma, Sachin Pawar, Sangameshwar Patil, Swapnil Hingmire, Girish K. Palshikar, and Pushpak Bhattacharyya. 2018. Identifying Participant Mentions and Resolving Their Coreferences in Legal Court Judgements. In *Text, Speech, and Dialogue*, pages 153–162, Brno, Czech Republic. Springer International Publishing.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. *CoRR*, abs/2111.09543. ArXiv: 2111.09543.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. *CoRR*, abs/1606.08415. ArXiv: 1606.08415.

Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022. Named Entity Recognition in Indian court judgments. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 184–193, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2016. Frustratingly Easy Neural Domain Adaptation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 387–396, Osaka, Japan. The COLING 2016 Organizing Committee.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, volume 1 of *ICML '01*, pages 282–289, Williamstown, MA, USA. Morgan Kaufmann Publishers Inc.

Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. 2019. Fine-grained Named Entity Recognition in Legal Documents. In *Semantic Systems. The Power of AI and Knowledge Graphs. Proceedings of the 15th International Conference (SEMANTiCS 2019)*, pages 272–287.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A Survey on Deep Learning for Named Entity Recognition. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Ashutosh Modi, Prathamesh Kalamkar, Saurabh Karn, Aman Tiwari, Abhinav Joshi, Sai Kiran Tanikella, Shouvik Guha, Sachin Malhan, and Vivek Raghavan. 2023. SemEval-2023 Task 6: LegalEval: Understanding Legal Texts. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Toronto, Canada. Association for Computational Linguistics (ACL).

Arttu Oksanen, Minna Tamper, Jouni Tuominen, Aki Hietanen, and Eero Hyvönen. 2022. A Tool for Pseudonymization of Textual Documents for Digital Humanities Research and Publication. In *Proceedings of the 6th Digital Humanities in the Nordic and Baltic Countries Conference (DHNB 2022)*, CEUR Workshop Proceedings, Germany. CEUR-WS.org.

Ildikó Pilán, Pierre Lison, Lilja Øvrelid, Anthi Papadopoulou, David Sánchez, and Montserrat Batet. 2022. The Text Anonymization Benchmark (TAB): A Dedicated Corpus and Evaluation Framework for Text Anonymization. *Computational Linguistics*, 48(4):1053–1101.

Vasile Păiș, Maria Mitrofan, Carol Luca Gasan, Vlad Coneschi, and Alexandru Ianov. 2021. Named Entity Recognition in the Romanian Legal Domain. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 9–18, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. Ontonotes: A large training corpus for enhanced processing. In Joseph Olive, Caitlin Christianson, and John McCary, editors, *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*. Springer, New York, NY, USA.

Lev Ratinov and Dan Roth. 2009. Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Tom Schamberger. 2021. Customizable Anonymization of German Legal Court Rulings using Domain-specific Named Entity Recognition. Master's thesis, Technical University Munich, Munich, Germany.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, Edmonton, Canada.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, \Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. 2019. Lookahead Optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.